

Um Tutorial sobre Análise de Componentes Principais para o Reconhecimento Automático de Faces

Relatório Técnico

Edson C. Kitani e Carlos E. Thomaz
Departamento de Engenharia Elétrica-Centro Universitário da FEI
Av. Humberto Alencar de Castelo Branco, 3972 SBC-SP-Brasil
{ekitani,cet}@fei.edu.br

RESUMO – O reconhecimento de faces é uma tarefa complexa sob o ponto de vista computacional. A enorme quantidade de informações fornecidas por uma imagem torna qualquer cálculo extremamente massivo sob o aspecto computacional e de ocupação de memória. Considere-se ainda que os objetos de interesse em uma imagem não possuem fundos e características comportadas. Logo, qualquer técnica para reconhecimento de faces deve levar em consideração esses dois aspectos, como obstáculos na obtenção de bons resultados nas análises. Este trabalho apresenta a técnica de análise de componentes principais (PCA) com aplicação na área de reconhecimento automático de faces, e fornece uma base teórica sobre esta técnica estatística para a compreensão do programa FACES.EXE desenvolvido pelos autores.

1 Introdução

Este trabalho apresenta um resumo sobre a pesquisa na área de reconhecimento de faces, concentrando-se na técnica estatística mais utilizada nesta aplicação, a Análise de Componentes Principais ou *Principal Components Analysis* (PCA). O relatório se baseará nos principais trabalhos relativos a esta técnica e ao longo do texto as referências indicarão trabalhos correlatos que complementam estas pesquisas. O objetivo deste trabalho é complementar a compreensão teórica do programa de reconhecimento de faces (FACES.EXE) desenvolvido pelos autores. A leitura deste texto não exige o leitor de consultar as referências, uma vez que aqui são apresentados apenas os resumos dos trabalhos de diversos autores.

Ao longo do desenvolvimento do texto, serão fornecidos trechos de pseudocódigos ou códigos em Matlab (MATLAB, 2001) relativos às etapas de cálculo, com o intuito de facilitar a compreensão, reduzir o tempo de pesquisa dos leitores, e favorecer o desenvolvimento de novos trabalhos a partir das informações fornecidas. Espera-se deste

modo reduzir o tempo de pesquisa das abordagens que já são consagradas e assim permitir uma concentração maior em outras técnicas estatísticas.

2 Representação de uma imagem discreta de face

Uma imagem discreta de face pode ser representada como uma matriz \mathbf{X} de dimensão $l \times c$, onde l é o número de linhas e c é o número de colunas da matriz. No entanto, é possível representar também essa mesma matriz como um vetor \mathbf{x}^T n -dimensional, onde $n = l.c$, e que na realidade representa a matriz \mathbf{X} concatenada. Se considerarmos que a imagem de uma face pode ser descrita como um vetor no espaço \mathcal{R}^n , então essa face é um ponto nesse espaço \mathcal{R}^n . Portanto um espaço de faces \mathbf{Zx} é um espaço multidimensional que contém todas as faces de um determinado conjunto de treinamento (TURK & PENTLAND, 1991). Isto é válido se considerarmos que trabalhamos em um sistema de coordenadas ortogonais e cujos vetores unitários $\varphi_1, \varphi_2, \dots, \varphi_n$ representam essa base, e assim um ponto nesse espaço pode ser escrito como uma combinação linear de vetores unitários ortonormais e das projeções do vetor \mathbf{x} de uma dada face no sistema de coordenadas ortogonais (CAROLI et al., 1982), isto é:

$$\mathbf{Zx}_i = \mathbf{x}_{i1}\varphi_1 + \mathbf{x}_{i2}\varphi_2 + \dots + \mathbf{x}_{in}\varphi_n, \quad (1)$$

onde i é uma face do espaço de faces \mathbf{Zx} .

Considere que temos agora um conjunto de N imagens discretas de faces. Se concatenarmos as imagens e as agruparmos em uma matriz \mathbf{Zx} , a matriz será formada por $N.l.c$ elementos. Deste modo, cada linha da matriz \mathbf{Zx} será a representação de uma face. A figura 1 ilustra didaticamente um espaço de faces, cuja dimensão é de N linhas por n colunas, onde N é o número de faces e n é o produto $l.c$ que representa a

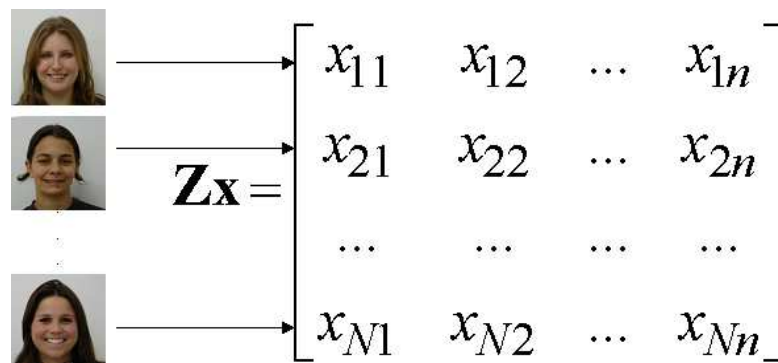


Figura 1. Representação do espaço de faces \mathbf{Zx} .

dimensionalidade desse espaço. Apesar das fotos da figura 1 serem coloridas, todo este trabalho foi desenvolvido com imagens monocromáticas, portanto cada célula da matriz Zx é representada dentro de uma escala de 0 a 255, que são as variações de tons de cinza em um sistema de representação de luminância com 8 bits de resolução.

Uma maneira de transformar uma matriz em um vetor linha é redimensionar essa matriz através do comando *reshape* do Matlab, como mostrado na figura 2. Para evitar um erro de interpretação, o código apresentado na figura 2 é exatamente da forma como deve ser digitado no Matlab. Mas para mantermos uma coesão matemática, devemos considerar que a concatenação transforma a matriz $X_{l \times c}$ em uma outra matriz $X_{l \times n}$.

```
// x é a matriz da imagem da face
// y é o vetor linha de x
y = reshape(x, (size(x, 1)*size(x, 2)));
```

Figura 2. Código para concatenar uma matriz de face.

A razão de se concatenar a matriz de face foi baseada no primeiro trabalho apresentado por (SIROVICH & KIRBY, 1987), onde eles apresentaram a hipótese da redução da dimensionalidade da matriz de face utilizando uma técnica de compressão de dados muito utilizada na área de processamento de sinais.

A técnica, que também é conhecida como Técnica de Expansão de Karhunen-Loève, normalmente é aplicada em sinais que variam no domínio do tempo e que são representados como vetores randômicos (FUKUNAGA, 1990). A figura 3 ilustra como

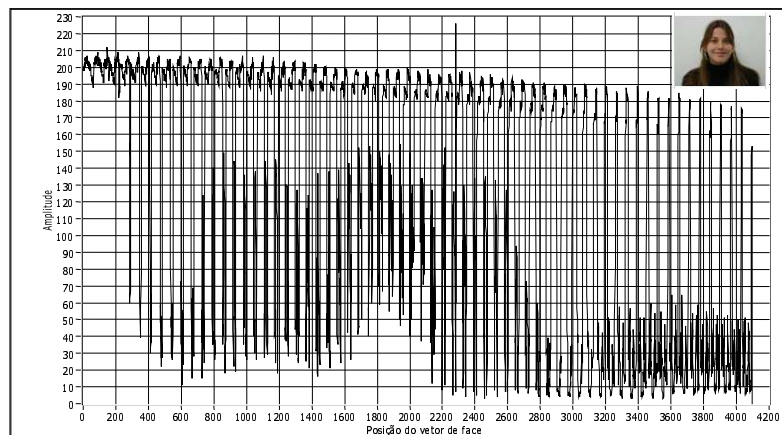


Figura 3. Gráfico do vetor de imagem de uma face. No canto superior direito da figura temos a imagem da face que é representada por este gráfico.

seria o vetor imagem de uma face visto como um sinal no domínio do tempo. Observe no gráfico da figura 3 que muita informação é redundante, e que a face poderia ser

representada somente pela variância que ocorre em torno de uma média (THOMAZ, 1999).

A partir desta hipótese construiu-se a tese de que uma imagem de face poderia ser representada com poucas componentes principais, pois muito da informação contida em uma imagem de face é redundante. Observa-se que o trabalho de (SIROVICH & KIRBY, 1987) estava baseado apenas na hipótese da representação de uma face em um espaço p – dimensional menor que o espaço original, onde p representa o número mínimo de componentes principais e consequentemente $p \ll n$.

Depois do estudo pioneiro de Sirovich e Kirby (SIROVICH & KIRBY, 1987), diversos trabalhos posteriores passaram então a utilizar o PCA para a caracterização das imagens de pessoas. O primeiro trabalho publicado utilizando o PCA para o reconhecimento de faces propriamente foi apresentado em 1991 por (TURK & PENTLAND, 1991). Hoje, este trabalho é um dos mais referenciados na área de reconhecimento de padrões e biometria. Existem diversos trabalhos que apontam para outras teorias sobre como se processa o reconhecimento de faces em seres humanos. Neste trabalho utilizamos as abordagens que são baseadas em características estatísticas das imagens ou *feature based*. O leitor interessado em conhecer outras abordagens deverá consultar inicialmente (CHELLAPPA et al., 1995), onde são descritas as várias abordagens de detecção e reconhecimento de faces.

Com o intuito de detalhar esse processo de caracterização de imagens por componentes principais, apresenta-se na figura 4 o gráfico de duas faces distintas. Observa-se no gráfico que alguns pontos são comuns, principalmente com relação ao fundo da imagem. Pode-se observar também que há uma densidade de variação em algumas regiões, o que pode significar que se capturarmos somente essas variações, poderemos representar cada face com um mínimo de informação, mas sem perda qualitativa da informação.

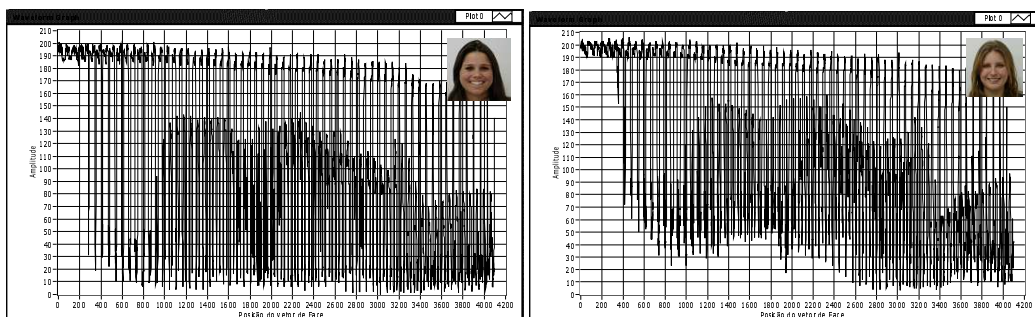


Figura 4. Gráfico de duas faces distintas.

Partindo-se da hipótese que há muita informação redundante e ruídos, e como foi dito anteriormente, se a variância de uma face pode ser descrita em torno de uma média, é razoável estudar essas variâncias em torno de uma média global. Tomemos a matriz do espaço de faces \mathbf{Zx} e calculemos o vetor média dessa matriz, isto é:

$$\bar{\mathbf{x}} = \left[\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \right], \quad (2)$$

onde N é o número total de faces do conjunto de treinamento.

O vetor médio resultante é conhecido como “face média”, e representa tudo aquilo que é comum a todas as faces. Um exemplo da imagem de uma “face média” pode ser visto

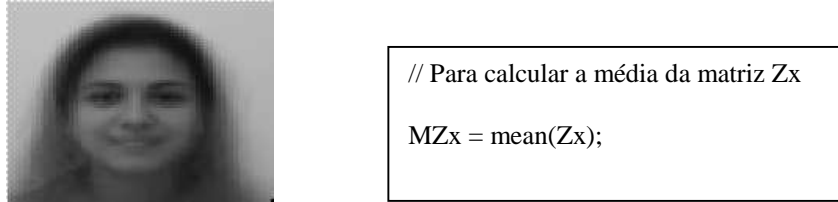


Figura 5. Imagem de uma face média.

na figura 5, ao lado encontra-se o código Matlab que calcula a média das colunas da matriz \mathbf{Zx} e salva o resultado em um vetor \mathbf{MZx} .

Observe, pela equação 2, que calcula-se a média aritmética de cada coluna, portanto o vetor média será de dimensão n . Esta é então a média global e cada face \mathbf{x}_i deverá variar em torno dessa média conforme:

$$\phi_i = \mathbf{x}_i - \bar{\mathbf{x}}, \quad (3)$$

onde $i = 1, 2, \dots, N$.

Este novo vetor ϕ_i contém todas as variações de uma determinada face i em torno da média $\bar{\mathbf{x}}$. Se então subtrairmos de todas as faces da matriz \mathbf{Zx} o vetor média $\bar{\mathbf{x}}$, teremos então uma nova matriz Φ , conforme a equação 4, que contém somente as variações de cada face em torno da média, e como consequência, uma matriz cuja média é zero e cuja a variância é máxima,

$$\Phi = [\phi_1, \phi_2, \dots, \phi_N]^T. \quad (4)$$

É interessante considerar que para efeito de cálculo, não podemos subtrair uma matriz de um vetor. Será necessário converter o vetor de média em uma matriz de mesma

dimensão que a matriz \mathbf{Zx} . O código da figura 6 ilustra uma forma de se criar uma matriz onde cada linha é o vetor média, e o número de linhas é igual a $N \times 1$.

```

// Criação da matriz média global e da matriz sem a média global

MZx = ones(size(x,1),1)*mean(x);
Zax = Zx-Mzx // Matriz Zx original sem a média global

```

Figura 6. Criação da matriz média global e da matriz Φ (THOMAZ, 1999).

O programa cria uma matriz coluna de ‘1’ com tamanho N e multiplica pela matriz linha da média global de tamanho $1 \times n$, resultando em uma matriz $\bar{\mathbf{x}}$ de tamanho $N \times n$.

3 Objetivo da redução da dimensionalidade dos dados

Até o momento foram discutidas a formação da matriz de faces e sua representação, mas não o objetivo geral da redução da dimensão e nem sua optimalidade.

Sabemos que as imagens discretas são representadas na forma matricial, e cujas resoluções podem variar muito, dependendo da qualidade da imagem que se deseja estudar. Porém em uma imagem discreta de 640×480 o total de células dessa matriz é equivalente a 307200. Logo, qualquer cálculo envolvendo dados dessa magnitude toma um esforço computacional elevado. Isto sem considerar o espaço de $N.n$ bytes de memória que ocuparia uma matriz de faces com essa resolução.

Assim fica clara a necessidade da redução da dimensionalidade, para que os dados possam ser manipulados e armazenados de forma mais eficiente. A questão agora é determinar uma transformação da matriz Φ de forma que a sua dimensão seja menor que essa nova matriz e carregue o máximo de informação possível sobre Φ .

Conforme descrito sucintamente no início da seção 2, uma face pode ser representada como uma combinação linear de vetores unitários no espaço \mathfrak{R}^n . Precisamos então encontrar um sistema de coordenadas nesse novo espaço que melhor represente todas as faces do conjunto Φ . Portanto cada face $\phi_i \in \Phi$ pode ser escrita como uma combinação linear de vetores unitários $\varphi_1, \varphi_2, \dots, \varphi_n$, tal que:

$$\phi_i = \sum_{j=1}^n y_j \varphi_j, \tag{5}$$

onde o vetor φ_j está sujeito a restrição de ortonormalidade, conforme a equação 6,

$$\varphi_i^T \varphi_j = \begin{cases} 1 & \text{para } i = j \\ 0 & \text{para } i \neq j \end{cases} \quad (6)$$

O vetor y_j é um novo vetor de características desse novo espaço, e são as projeções sobre os eixos de cada vetor φ_j . Se cada face ϕ_i é representado pela equação 3, e sendo esta uma combinação linear, podemos expandi-la da seguinte forma sem qualquer erro:

$$\phi_i = \sum_{j=1}^p y_j \varphi_j + \sum_{j=p+1}^n y_j \varphi_j, \quad (7)$$

onde $p < n$.

Considere agora que no segundo termo da equação 7 as variáveis y possam ser substituídas por constantes b_j , então este novo vetor ϕ_i^* , onde

$$\phi_i^* = \sum_{j=1}^p y_j \varphi_j + \sum_{j=p+1}^n b_j \varphi_j, \quad (8)$$

apresentará um erro ε em relação ao vetor original ϕ_i , e o vetor erro ε_i poderá ser determinado pela seguinte expressão:

$$\varepsilon_i = \phi_i - \phi_i^*. \quad (9)$$

O vetor erro ε_i representa o desvio de reconstrução da face ϕ_i^* em relação a face de origem ϕ_i e o objetivo é minimizar o vetor erro ε_i enquanto se tenta minimizar também o número de componentes principais que formam a face ϕ_i^* .

Trabalhando com a equação 9, vamos substituir ϕ_i e ϕ_i^* pelas suas combinações lineares, isto é.

$$\varepsilon_i = \left[\sum_{j=1}^p y_j \varphi_j + \sum_{j=p+1}^n y_j \varphi_j \right] - \left[\sum_{j=1}^p y_j \varphi_j + \sum_{j=p+1}^n b_j \varphi_j \right]. \quad (10)$$

Como descrito por (FUKUNAGA, 1990), as componentes escalares $(y_{p+1}, \dots, y_n)^T$ são componentes que estaríamos desconsiderando, uma vez que, por hipótese, estaríamos eliminando-as, pois $p \ll n$. Logo, essas componentes poderiam ser substituídas por constantes escalares de modo a manter um vetor de mesma dimensão que ϕ_i , e assim

representaríamos cada face com somente p componentes como é mostrado por (FUKUNAGA, 1990), onde:

$$\varepsilon_i = \sum_{j=1}^n y_j \cdot \varphi_j - \left(\sum_{j=1}^p y_j \cdot \varphi_j + \sum_{j=p+1}^n b_j \cdot \varphi_j \right). \quad (11)$$

Observa-se na equação 11 que o primeiro e segundo termos têm os primeiros p termos em comum, ou seja:

$$\varepsilon_i = \sum_{j=1}^n y_j \cdot \varphi_j - \sum_{j=1}^p y_j \varphi_j - \sum_{j=p+1}^n b_j \varphi_j. \quad (12)$$

Logo, eliminando-se esses termos em comum, resta:

$$\varepsilon_i = \sum_{j=p+1}^n y_j \varphi_j - \sum_{j=p+1}^n b_j \varphi_j. \quad (13)$$

Colocando-se φ_j em evidência

$$\varepsilon_i = \sum_{j=p+1}^n (y_j - b_j) \varphi_j, \quad (14)$$

podemos concluir que o vetor de erro depende exclusivamente das componentes principais que estaríamos descartando, e para se determinar a magnitude desse erro, deve-se determinar o erro médio quadrático do vetor de erro. Como desejamos que essa magnitude seja a mínima possível, devemos procurar uma base para os vetores y_{p+1}, \dots, y_n e valores para as constantes b_j que minimizem esse erro (OSUMA, 2004).

Como estamos trabalhando com vetores que são aleatórios, existe um conceito estatístico de esperança ou expectativa matemática para uma variável aleatória, que é definido da seguinte forma:

$$E(X) = x_1 P(X = x_1) + \dots + x_n P(X = x_n) = \sum_{i=1}^n x_i P(X = x_i). \quad (15)$$

O que significa dizer que existe uma probabilidade da variável \mathbf{X} assumir os valores x_1, x_2, \dots, x_n , dado a probabilidade deles ocorrerem. Caso a probabilidade $P(X = x_i)$

seja igual para todas as variáveis x_i , então a esperança $E(X)$ representará a média da ocorrência de todas as variáveis, como indicado na equação 16 (SPIEGEL, 2004) abaixo

$$E(X) = \frac{x_1 + \dots + x_n}{n}. \quad (16)$$

Portanto, (FUKUNAGA, 1990) demonstra que o erro médio quadrático da equação 14, considerando-se as $n - p$ características eliminadas, pode ser escrito como :

$$\varepsilon^2(P) = E \left\{ \left\| \sum_{j=p+1}^n (y_j - b_j) \varphi_j \right\|^2 \right\}. \quad (17)$$

Expandindo a equação 17, temos:

$$\varepsilon^2(P) = E \left[\sum_{k=p+1}^n \sum_{j=p+1}^n (y_k - b_k)(y_j - b_j) \varphi_k^T \varphi_j \right] = \sum_{j=p+1}^n E(y_j - b_j)^2. \quad (18)$$

Na equação 18, como φ_j é um vetor ortonormal, o produto dele e sua transposta será igual a 1, e os índices k e j que aparecem na equação expandida decorre do fato de que teremos uma seqüência de somas dos produtos dos termos $(y_j - b_j)$, assim teremos dois índices para distinguir cada termo. A equação 18 segue a propriedade que o valor esperado da soma é igual a soma dos valores esperados.

Resta-nos agora determinar o que minimizaria o erro ε . FUKUNAGA (FUKUNAGA, 1990) propõe que uma forma de minimização seria derivar parcialmente a equação 18 em relação a b_j , de modo que ela fosse igual a zero, ou seja

$$\frac{\partial}{\partial b_j} E[(y_j - b_j)^2] = 0, \quad (19)$$

onde $j = p + 1, \dots, n$

$$-2(E[y_j] - b_j) = 0 \quad (20)$$

$$b_j = (E[y_j]). \quad (21)$$

Se uma face ϕ_i pode ser representada como uma combinação linear de vetores como foi mostrado na equação 5, então podemos escrever a equação 5 como um produto vetorial, como mostrado na equação 22 abaixo

$$\Phi = \Gamma \mathbf{Y}, \quad (22)$$

onde $Y = [y_1, y_2, \dots, y_n]^T$, $\Gamma = [\phi_1, \phi_2, \dots, \phi_n]$, e $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$.

Deste modo, o vetor \mathbf{Y} pode ser escrito como uma combinação linear do vetor Φ e do vetor ortonormal Γ , tal que

$$y_j = \phi_j^T \phi_j, \quad (23)$$

onde $j = 1, 2, 3, \dots, n$.

Substituindo a equação 23 na equação 21, temos

$$b_j = (E[\phi_j^T \phi_j]) \quad (24)$$

$$b_j = \phi_j^T E(\phi_j), \quad (25)$$

isto porque a expectativa do vetor ortonormal é ele próprio.

Uma solução trivial seria que os valores de b_j fossem exatamente iguais a y_j , mas o que procuramos é uma solução ótima e não a trivial, e considere-se ainda que y_j é um vetor aleatório, portanto $E(y_j)$ é uma expectativa matemática.

Substituindo essa expectativa $E(y_j)$ no lugar de b_j , tem-se para a equação do erro quadrático

$$(\bar{\varepsilon})^2 = \sum_{j=p+1}^n E[(y_j - E[y_j])^2]. \quad (26)$$

Conforme visto anteriormente, o vetor y_j pode ser representado como um produto dos vetores ortonormais e das componentes da face que queremos eliminar, e podemos escrever que

$$y_j = \phi_j^T \phi_j, \quad (27)$$

para $j = p + 1, \dots, n$.

Substituindo a equação 27 na equação 26:

$$\overline{(\varepsilon)}^2 = \sum_{j=p+1}^n E \left[\left(\varphi_j^T \phi_j - E[\varphi_j^T \phi_j] \right)^2 \right]. \quad (28)$$

Como descrito por (FUKUNGA, 1990), quando um vetor \mathbf{X} é transformado linearmente em um outro vetor n -dimensional \mathbf{Y} , o vetor \mathbf{Y} pode ser descrito em função de \mathbf{X} , como pode ser visto na equação 29 abaixo

$$\mathbf{Y} = \mathbf{A}^T \mathbf{X}, \quad (29)$$

onde a matriz \mathbf{A} é uma matriz de dimensão $n \times n$. Desta forma, é possível escrever que:

$$\overline{\mathbf{Y}} = E(\mathbf{Y}) = \mathbf{A}^T E(\mathbf{X}) = \mathbf{A}^T \overline{\mathbf{X}}, \quad (30)$$

e assim a matriz de covariância $\Sigma_{\mathbf{Y}}$ pode ser determinada por:

$$\Sigma_{\mathbf{Y}} = E[(\mathbf{Y} - \overline{\mathbf{Y}})(\mathbf{Y} - \overline{\mathbf{Y}})^T]. \quad (31)$$

Substituindo o vetor \mathbf{Y} pela transformação linear \mathbf{X} , conforme a equação 26, temos:

$$\Sigma_{\mathbf{Y}} = E[(\mathbf{A}^T \mathbf{X} - \mathbf{A}^T E(\mathbf{X}))(\mathbf{A}^T \mathbf{X} - \mathbf{A}^T E(\mathbf{X}))^T]. \quad (32)$$

Colocando a matriz \mathbf{A} em evidencia, temos:

$$\Sigma_{\mathbf{Y}} = \mathbf{A}^T E[(\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T] \mathbf{A}. \quad (33)$$

Retornando a equação 28 e expandindo-a, temos:

$$\overline{(\varepsilon)}^2 = \sum_{j=p+1}^n E \left[\left(\varphi_j^T \phi_j - E[\varphi_j^T \phi_j] \right) \left(\varphi_j^T \phi_j - E[\varphi_j^T \phi_j] \right)^T \right], \quad (34)$$

e de forma análoga, podemos escrever a equação 34 conforme a equação 33, ou seja,

$$\overline{(\varepsilon)}^2 = \sum_{j=p+1}^n \varphi_j^T E \left[\left(\phi_j - E[\phi_j] \right) \left(\phi_j - E[\phi_j] \right)^T \right] \varphi_j. \quad (35)$$

Como o produto dos termos interno ao colchete representa uma matriz de covariância de ϕ_j , podemos rescrever 35 como:

$$(\bar{\varepsilon})^2 = \sum_{j=p+1}^n \varphi_j^T \Sigma \varphi_j. \quad (36)$$

FUKUNAGA (FUKUNAGA, 1990) mostra que o valor ótimo para o vetor φ_j deve satisfazer a relação abaixo:

$$\Sigma \varphi_j = \lambda_j \varphi_j, \quad (37)$$

onde

$$\lambda_j = \varphi_j^T \Sigma \varphi_j, \quad (38)$$

e (φ_j, λ_j) são os pares de autovetores e autovalores da matriz de covariância Σ .

Substituindo a equação 38 na equação 36, temos:

$$(\bar{\varepsilon})^2 = \sum_{j=p+1}^n \lambda_j. \quad (39)$$

Como λ_j são os autovalores da matriz de covariância Σ , se eliminarmos os menores autovalores minimizaremos o erro ε .

Em (FUKUNAGA, 1990), o leitor encontrará uma demonstração completa da Expansão Discreta de Karhunen-Loève, aplicado para um vetor aleatório. Foi apresentada nesta seção uma adaptação daquela prova para o modelo de faces descrito por (SIROVICH & KIRBY, 1987) e por (TURK & PENTLAND, 1991), complementado e adaptado pelos trabalhos apresentados por (ROMDHANI, 1996) e (OSUMA, 2004).

4 Cálculo dos Autovetores e Autovalores de Φ

Na seção anterior foi discutido a motivação e a prova da redução da dimensionalidade da matriz Φ . Nesta seção veremos como calcular os autovetores dessa matriz de face de modo a encontrar os autovalores correspondentes, que são os menores e assim remover as componentes que não trazem informação relevante sobre a variabilidade da amostra.

4.1 Autovetores e autovalores

Em problemas de álgebra linear é freqüente o aparecimento de questões sobre como encontrar parâmetros que satisfaçam equações do tipo:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}, \quad (40)$$

onde \mathbf{A} é uma matriz quadrada, \mathbf{x} um vetor e λ um escalar. Uma solução para o problema seria $\mathbf{x} = \mathbf{0}$, mas essa solução trivial não traz nenhuma informação relevante, portanto, devem existir outros valores de \mathbf{x} e λ que satisfazem a equação 40.

Rescrevendo a equação 40 de uma outra maneira, podemos multiplicar o termo da direita pela matriz identidade sem que isto modifique a igualdade, ou seja,

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{I}\mathbf{x}. \quad (41)$$

Agora modifiquemos a equação 41 para igualar essa equação a zero, isto é

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}. \quad (42)$$

Se a matriz \mathbf{A} for de dimensão $n \times n$, teremos um conjunto de n equações lineares com n incógnitas.

$$\begin{bmatrix} a_{11} - \lambda_1 & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda_2 & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & \dots & \dots & a_{nn} - \lambda_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{bmatrix} = \mathbf{0}. \quad (43)$$

Uma solução para a equação 43, seria fazer o determinante da matriz ser igual a zero, ou seja, $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. Essa equação, comumente chamada de equação característica da matriz \mathbf{A} , produz um polinômio de grau n em λ . Esse polinômio tem então n raízes reais ou complexas $(\lambda_1, \lambda_2, \dots, \lambda_n)$, e para cada raiz λ_j obteremos valores x_j que irão compor o autovetor de \mathbf{x} .

O leitor interessado em compreender a solução para esse problema mais detalhadamente deve consultar (PETTOFREZZO, 1966), (BARROSO et al., 1987), (SALVETTI, 1982), (CALLIOLI et al., 1978) e (HADLEY, 1961).

4.2 Calculando autovalores e autovetores

Vimos na seção 3 que os autovalores e autovetores são calculados a partir da matriz de covariância do espaço Φ . Porém calcular a matriz de covariância Σ diretamente de Φ exigirá um esforço computacional enorme, visto que o produto de $\Phi^T \cdot \Phi$ produzirá uma matriz de 4096×4096 para uma imagem de resolução 64×64 , por exemplo. Portanto como apresentado em (THOMAZ, 1999), a computação da matriz $\Phi \cdot \Phi^T$ exigirá um esforço muito menor, produzindo uma matriz de dimensão $N \times N$. Thomaz (THOMAZ, 1999) mostra que a solução para a equação 43 é satisfeita pela computação da matriz $\Phi^T \Phi$.

Como ainda trabalhamos com amostras de variáveis aleatórias e não com toda a população, (FUKUNAGA, 1990) demonstra que a matriz de covariância de $\Sigma\Phi$ deve ser calculada pela equação 44 abaixo.

$$\Sigma\Phi = \left(\frac{\Phi \cdot \Phi^T}{N-1} \right). \quad (44)$$

O problema é que a matriz $\Sigma\Phi$ de dimensão $N \times N$ não determina os autovetores Ψ para os autovalores Λ que são zero, isto porque um número de $(n-p)$ elementos são suprimidos. Isto faz com que os autovetores gerados por essa matriz sejam ortogonais entre si, mas não ortonormais. FUKUNAGA (FUKUNAGA, 1990) propõe uma solução para esse problema, recalculando os autovetores conforme indicado na equação 45 abaixo

$$\Psi^* = \frac{\Phi^T \cdot \Psi \cdot (\Lambda)^{-\frac{1}{2}}}{\sqrt{N-1}}, \quad (45)$$

onde Ψ^* será a nova matriz de autovetores que é ortogonal e ortonormal. Na figura 7 encontramos o código em Matlab para calcularmos a matriz de covariância de Φ , os autovetores e os autovalores, e lembrando que as variáveis utilizadas no código diferem um pouco do texto, para não coincidir com os mneumônicos do Matlab. A matriz **Pt** criada pelo processamento do programa da figura 7 é uma matriz cujos os autovalores não estão ordenados conforme a ordem decrescente dos autovalores. O próximo trecho de programa apresentado na figura 8 fará a ordenação decrescente do vetor de autovalores e depois ordenará os autovetores conforme essa nova ordem.

```

// Zax é a matriz de faces sem a média global
V = Zax*Zax' ; // Em V temos uma matriz de dimensão k
(Qv, Qa) = eig(V, 'nobalance'); // Qv é o autovetor e Qa o autovalor
Pt = (Zax'*Qv*(Qa^(0,5)))/(sqrt(m-1));
// Pt ;e equivalente ao  $\Psi^*$  do texto

```

Figura 7. Código para calcular a matriz de covariância Σ e os respectivos autovalores e autovetores.

Explicando melhor, cada coluna da matriz \mathbf{Pt} representa uma componente principal da matriz de faces, e o autovalor associado a esse autovetor mostra a representatividade dessa componente principal em relação ao conjunto de autovetores. Desta forma, o índice de coluna do autovetor é o mesmo do índice de coluna do autovalor. Assim, se movemos o autovetor de coluna 3 para a coluna 1, devemos também mover o autovalor de coluna 3 para a coluna 1. A figura 8 mostra o trecho do programa que faz a ordenação dos autovalores e depois dos autovetores. A matriz \mathbf{P} contém agora os autovetores ordenados na ordem definida pelos autovalores, e olhando o vetor de autovalores podemos determinar a representatividade de cada autovetor na caracterização de uma face.

```

// Ordena em ordem decrescente os autovetores
(Ks, Ki) = sort(diag(Qa));
Ki = flipud(Ki);
P=Pt(:,Ki(1:n));
K = flipud(Ks);
K=K(1:n);

```

Figura 8. Programa que executa a ordenação e montagem da matriz de autovetores conforme os autovalores.

A equação 46 representa esse cálculo da porcentagem da representatividade de cada autovetor na formação das faces do espaço Φ , e denominaremos esse vetor de $P(\lambda_i)$, tal que

$$P(\lambda_i) = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} \cdot 100\% , \quad (46)$$

onde $i = 1, 2, \dots, p$.

Somam-se todos os autovalores, mesmo aqueles que serão descartados, e depois calcula-se a razão entre cada autovalor e a soma total. Retornando à matriz \mathbf{P} calculada

no programa da figura 8, chamaremos essa matriz \mathbf{P} de matriz \mathbf{P}_{pca} de modo a indicar que essa matriz, de dimensão $N \times p$, representa as faces \mathbf{X} nesse novo subespaço, de menor dimensão. Portanto, toda a análise será executada a partir dessa matriz.

Sob o aspecto da redução da dimensionalidade dos dados, o processo se encerra aqui, mas como desejamos comparar uma face de teste com o conjunto de treinamento, temos mais alguns passos a seguir.

4.3 Face de teste projetado no subespaço PCA

Como foi descrito no tópico 4.2, criamos um novo subespaço vetorial de dimensão $N \times p$, onde agora cada face é representada por uma combinação linear de vetores com uma dimensão bem menor. O problema é que quando formos executar uma comparação com uma face que não pertence ao conjunto de treinamento, devemos fazê-lo neste subespaço. Portanto a nova face deverá ser projetada nesse espaço, e isto é conseguido multiplicando-se a matriz da face de teste pela matriz de transformação Ψ^* , como descrito a seguir.

Seja \mathbf{x}_{teste} a matriz de uma face do conjunto de teste; então o vetor \mathbf{x}_{teste}^T será essa matriz de face \mathbf{x}_{teste} concatenada, e o vetor ϕ_{teste}^T será o vetor dessa face de teste concatenado e subtraído da média global $\bar{\mathbf{x}}$, como visto na equação 44 abaixo.

$$\phi_{teste}^T = \mathbf{x}_{teste} - \bar{\mathbf{x}}. \quad (44)$$

Esta nova matriz linha será então multiplicada pela matriz de transformação Ψ^* , produzindo uma nova matriz linha de dimensão $1 \times p$. Chamaremos esta nova matriz linha de ϕ_{pca} indicando que é uma face de teste projetada no espaço PCA.

$$\phi_{pca} = \phi_{teste}^T \cdot \Psi^*. \quad (45)$$

Com a face de teste projetada no espaço Ψ^* a questão agora é determinar uma regra de decisão para se testar a similaridade dessa face de teste com todas as faces da matriz de treinamento. A regra de decisão utilizada no programa FACES.EXE foi uma técnica de classificação por distância, onde o vetor ϕ_{pca} de teste foi subtraído de cada face da matriz Ψ^* , conforme a equação 46 abaixo

$$\delta_i = \phi_i - \phi_{pca}, \quad (46)$$

para $i = 1, 2, \dots, N$.

O novo vetor δ_i representa o erro entre a face de teste ϕ_{pca} e cada face ϕ_i do conjunto de treinamento. Portanto, busca-se a face que minimiza δ_i , e como δ é um vetor, calculou-se o valor médio quadrático (RMS *Root Mean Square*) de cada vetor de erro δ_i . O vetor que apresentar o menor valor RMS indica que a diferença $\delta_i - \delta_{pca}$ é a menor, e portanto é a de maior similaridade entre a face de teste ϕ_{pca} e a face de treinamento ϕ_i onde i é o índice do vetor de erro δ_i que apresentou o menor erro. A equação 47 mostra o cálculo do valor RMS para o vetor de erro δ_i .

$$RMS\delta_i = \left(\sqrt{\frac{1}{P} \sum_{j=1}^P \delta_j^2} \right), \quad (47)$$

onde $i = 1, 2, \dots, N$.

Agrupando as equações 46 e 47, temos:

$$RMS\delta_i = \left(\sqrt{\frac{1}{P} \sum_{j=1}^P (\phi_j - \phi_{pca})^2} \right), \quad (48)$$

para $i = 1, 2, \dots, N$.

A figura 9 mostra um típico gráfico do vetor de erro δ_i , onde observa-se que há um erro grande entre as primeiras componentes principais e uma diminuição no erro nas últimas componentes. Isto ocorre porque os autovetores são ordenados conforme a sua representatividade em relação a cada face. Como o vetor de erro não é normalizado, a amplitude do erro fornece a falsa impressão de que o erro diminui.

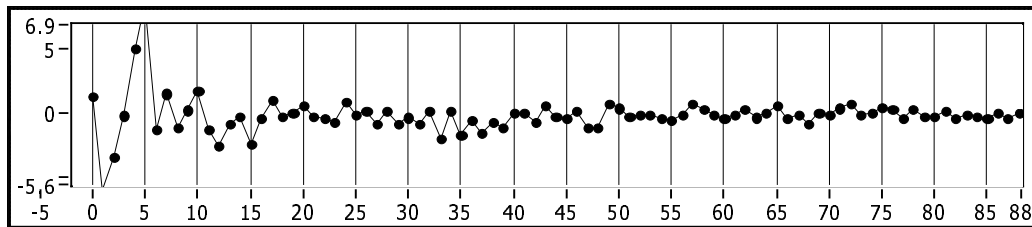


Figura 9. Gráfico de erro δ_i entre duas faces. O eixo x representa as componentes principais e o eixo y o valor absoluto do erro $(\phi_j - \phi_{pca})$.

4.4 Banco de Faces da FEI.

O programa FACES.EXE foi testado com o banco de faces da FEI, que é composto atualmente (Março/2006) de fotos de 102 voluntários, formados por professores, funcionários e alunos da FEI, em 12 diferentes poses, que variam com fotos de perfil esquerdo para o perfil direito. As imagens utilizadas para este teste, também não sofreram nenhuma normalização geométrica ou equalização de histograma. As imagens do banco de faces da FEI são coloridas e com resolução de 640×480 pixels, mas para se reduzir o esforço computacional, o conjunto de treinamento e de testes foi reduzido na sua resolução em 64×64 pixels e tornadas monocromáticas. Ainda, por uma simples conveniência, e para se obter uma ampla variação de características, foram utilizadas somente as faces femininas do banco de faces. Como as mulheres apresentam diferentes biotipos e aparências bem distintas (tipo de cabelo, adornos, etc.), optou-se por formar um conjunto de treinamento somente com essas faces. No total foram utilizadas 34 fotos em 10 diferentes poses. A figura 10 apresenta uma amostra do conjunto de treinamento do banco de faces e a figura 11 um número de imagens desse conjunto, mas com a resolução reduzida e tornadas monocromáticas.



Figura 10. Amostra do conjunto de faces do banco de faces da FEI.



Figura 11. Amostra do conjunto de treinamento, mas reduzidas na resolução e tornadas monocromáticas.

5 Experimentos e resultados

No experimento foram utilizadas todas as faces femininas do banco de faces da FEI composto de 34 pessoas em 12 poses ou expressões diferentes. Para o primeiro teste montou-se um grupo de treinamento formado por 10 faces em 7 diferentes poses, como pode ser visto no exemplo da figura 12. Os resultados foram avaliados através da distância Euclidiana entre o vetor de uma face de teste transformada no espaço PCA e com todas as faces do conjunto de treinamento Ψ^* .



Figura 12 Amostra do conjunto de treinamento número 1.

Este primeiro grupo de treinamento foi formado com a seqüência de fotos, de índice número 1, de 10 pessoas, e seguidas das fotos de índice número 2, 3, 11, 8, 9, e 10, totalizando 70 fotos. Cada foto do banco de faces da FEI possui um nome de arquivo, que é formado da seguinte maneira: seja aaa o número da pessoa no banco de faces, bb a pose da pessoa, então um nome de arquivo é formado pela composição do número da pessoa no banco e sua pose, mais a extensão do arquivo. Como pode ser visto na figura 12, a foto de uma pessoa é referenciada pelo seu nome de arquivo (aaa-bb.jpg) indicando sua posição no banco de faces (aaa) e a pose (bb). Restaram, portanto as fotos de índice bb 4, 5, 6, 7 e 12 de cada uma dessas faces, e todas as outras que não pertencem ao conjunto de treinamento.

No trabalho apresentado por (SIROVICH & KIRBY, 1989) o conjunto de treinamento foi formado por faces masculinas frontais, alinhadas, normalizadas e com a mesma escala. A reconstrução das faces foi considerada adequada com a utilização de apenas 40 componentes principais.

O conjunto de treinamento utilizado neste trabalho foi montado sem nenhuma normalização, alinhamento ou ajuste de escala, e com variações de pose. O módulo PCA foi ajustado para trabalhar com 60 componentes principais, pois com esta quantidade de componentes principais, a reconstrução das faces se mostrou mais adequada para descrever todas as variações das faces do conjunto de treinamento.

Depois da formação da matriz Zx do conjunto de treinamento, para os testes foram apresentados todas as fotos de índice 4 e os resultados avaliados manualmente com relação ao acerto ou erro. Em seguida todas as fotos de número 5, e assim sucessivamente, totalizando 20 fotos com pose para a esquerda e 20 fotos com pose para direita até finalizar com as 10 fotos frontais de índice 12. Na figura 13 temos uma amostra do conjunto de teste.



Figura 13. Amostra do conjunto de teste.

Quando o programa FACES.EXE indicava uma face errada, não necessariamente a pessoa retornada pelo PCA estava errada, mas sim a sua pose. Observa-se que no caso das imagens frontais, a baixa taxa de acerto em relação às variações de pose para esquerda e direita, deve-se principalmente ao baixo número de exemplos de imagens de cada pessoa nesta pose, além do que a foto apresentada não tinha expressão neutra. A tabela 1 mostra o resultado do teste somente com o módulo PCA, onde o número em parênteses na coluna da esquerda representa a quantidade de faces de testes apresentadas ao programa.

Tabela 1. Porcentagem de acerto

POSE	PCA
Para Esquerda (20)	95%
Para Direita (20)	95%
Frontal (10)	80%
Total geral (50)	94%

A taxa de acerto foi calculada conforme a equação 49 abaixo,

$$Taxa = \frac{NF}{TF} \cdot 100\%, \quad (49)$$

onde NF é número de faces corretas que foram reconstruídas pelo PCA quando apresentados ao conjunto de teste, e TF é o total de faces da pose em teste, ou seja, esquerda, direita e frontal. Com relação aos erros apresentados pelo PCA no reconhecimento da pose, deve-se principalmente ao fato de que as poses não coincidiam com a pose definida pelo número do arquivo. O leitor interessado em mais detalhes sobre o resultados desses testes, deve consultar (KITANI & THOMAZ, 2005).

6 Conclusão

Neste tutorial foram apresentadas as bases teóricas para a compreensão da técnica estatística de análise de componentes principais (PCA), que é muito utilizada nas aplicações de caracterização e reconhecimento de faces. Esta base teórica permite a compreensão do funcionamento do programa FACES.EXE no módulo relativo ao PCA. Foram também abordadas as bases matemáticas do PCA bem como os algoritmos necessários para a implementação de um aplicativo que trabalhe com PCA, não importando necessariamente a plataforma de programação. Os experimentos preliminares que utilizaram a técnica PCA para o reconhecimento da pose das faces do banco de dados da FEI apresentaram resultados de classificação satisfatórios, comprovando a potencialidade da técnica para o reconhecimento de faces.

7 Referências

BARROSO, Leônidas C. et al., **Cálculo Numérico com Aplicações**, 2^a ed., São Paulo, SP: Editora Habra, 1976.

CALLIOLI, Carlos R., et al. **Álgebra Linear e Aplicações**, 2^a ed., São Paulo, SP: Editora Atual, 1978.

CHELLAPPA, C. L., et al., **Human and Machine Recognition: A Survey**, Proceedings of the IEEE, vol 83, no. 5, pp 705-740, May 1995.

FUKUNAGA, Keinosuke, **Introduction to Statistical Pattern Recognition**, 2^a ed., Boston, MA: Academic Press, 1990.

HADLEY, G., **Álgebra Linear**, Rio de Janeiro, RJ: Editora Forense-Universitária, 1979.

KITANI, Edson C., THOMAZ, Carlos E., **Abordagem PCA+LDA para Reconhecimento de Faces com Variação de Pose**, Relatório do Seminário de Visão Computacional do Mestrado da FEI, 7 pg., FEI, Nov. 2005.

MATLAB, **Getting Started with Matlab 6**, Mathworks Inc, 2001.

OSUMA, Ricardo Gutierrez, **Principal Components Analysis, Lecture Notes 9**, Texas A&M University, Texas, 2004, disponível em www.courses-cs.tamu.edu/rgutier/cs790-w02, acessado em 20/12/2005.

PETTOFREZZO, Anthony J., **Matrices and Transformations**, New York, NY: Dover Science, 1966.

RONDHANI, Sami, **Face Recognition using Principal Components Analysis**, Master These, 86 pg., 1996.

SALVETTI, Dirceu Douglas, **Tópicos de Cálculo Numérico**, São Bernardo do Campo, SP: Editora FCA-FEI, 1982.

SIROVICH, L. KIRBY, M, **Low-dimensional Procedure for the Characterization of Human Faces**, Journal of the Optical Society of America, vol. 4, pp 519-524, 1987.

SPIEGEL, Murray R., **Probabilidade e Estatística**, São Paulo, SP: Editora Pearson Education, 2004.

THOMAZ, Carlos Eduardo, **Estudo de Classificadores para Reconhecimento Automático de Faces**, Tese de Mestrado, 104 pg., PUC RJ, 1999.

TURK, M., PENTLAND, A., **Eigenfaces for Recognition**, Journal of Cognitive Neuroscience, MIT, vol. 73, pp. 71-86, 1991.